Experimentierkasten

Aufgaben

1 Ein Hersteller von Experimentierkästen für Kinder und Jugendliche benötigt eine Buchungssoftware, damit Kundinnen und Kunden (im Folgenden Kunden genannt) auch online Experimentierkästen und einzelne Bauteile bestellen können. Teile dieser Software sind von Ihnen anzufertigen.

1.1 Jeder Kunde kann sich passende Experimentierkästen des Herstellers anzeigen lassen, wobei der Kunde dafür das empfohlene Mindestalter und eine Preisobergrenze auswählen muss. Auch kann sich jeder Kunde eine Liste mit allen Bauteilen zu einem Experimentierkasten anzeigen lassen. In diesem Fall muss der Kunde den betreffenden Kasten vorab auswählen. Ebenfalls kann der Kunde einzelne Bauteile (z. B. als Ersatz) bestellen. Dazu wählt der Kunde die entsprechenden Bauteile aus und verändert die gewünschte Anzahl, falls er mehr als ein Exemplar eines bestimmten Bauteils bestellen möchte.
Entwickeln Sie ein aussagekräftiges Use-Case-Diagramm (Anwendungsfalldiagramm). Bestimmen Sie dabei die menschlichen Akteure, die Beziehungen zwischen Akteuren und Anwendungsfällen und eventuell vorhandene include- bzw. extend-Beziehungen.

(8 BE)

1.2 Ein Teil des UML-Klassendiagramms der Buchungssoftware wurde bereits erstellt (Material 1).

1.2.1 Erläutern Sie den Zusammenhang zwischen den Klassen `Experimentierkastenliste` und `Experimentierkasten` und erklären Sie die Funktion des Attributs `instance`.

(4 BE)

1.2.2 Beschreiben Sie die Art und die Multiplizitäten der Assoziation zwischen den Klassen `Bestellung` und `Bestellposition` und erklären Sie die Funktion der Klasse `Bestellposition`.

(5 BE)

1.2.3 Jede Bestellposition bezieht sich auf genau ein Bauteil. Das Bauteil kennt jedoch nicht seine möglichen Bestellpositionen. Jedes Bauteil hat eine Teilenummer, eine Bezeichnung und einen Preis. Jedes Bauteil befindet sich in mindestens einem Experimentierkasten des Herstellers. Ein Bauteil kann auch mehrmals in einem Experimentierkasten enthalten sein. Ein Bauteil kennt jedoch nicht die Experimentierkästen, in denen es enthalten ist.
Ergänzen Sie das UML-Klassendiagramm (Material 1) mithilfe der Informationen aus diesem Aufgabenteil. Modellieren Sie dazu sämtliche Attribute (sinnvolle Datentypen), einen zweckmäßigen Konstruktor und die Assoziationen samt Assoziationsnamen, Multiplizitäten und Navigierbarkeit.

Hinweis: Getter- und Setter-Methoden sind nicht zu modellieren.

(6 BE)

1.3 Die Methode `experimentierkastenAnzeigen(…)` der Klasse `Experimentierkastenliste` liefert eine Liste mit allen Experimentierkästen zurück, deren empfohlenes Mindestalter kleiner gleich dem übergebenen Alter ist und deren Preis nicht größer als das 1,1-fache des übergebenen Preises ist. Die Methode `bauteileAnzeigen(…)` der Klasse `Experimentierkastenliste` liefert eine Liste mit allen Bauteilen des übergebenen Experimentierkastens zurück. Der Wert des Attributs `artikelnr` der Klasse `Experimentierkasten` soll fortlaufend (beginnend mit 1) gesetzt werden.
Implementieren Sie die Klassen `Experimentierkastenliste` und `Experimentierkasten` entsprechend Ihres ergänzten UML-Klassendiagramms aus Aufgabe 1.2.3 sowie der Aufgabenstellung in Aufgabe 1.2.3 und entsprechend der obigen Beschreibung in der von Ihnen im Unterricht verwendeten objektorientierten Programmiersprache.

(18 BE)

1.4 Mithilfe der Methode `bauteileBestellen(…)` der Klasse `Kunde` kann ein Kunde ein oder mehrere Bauteile bestellen. Über das Array `teile` werden diese an die Methode übergeben. Die Anzahl, die ein Kunde von jedem Bauteil bestellt, wird über das Array `anzahl` an die Methode übergeben. Dabei entspricht die Indexposition eines Bauteils im Array `teile` der Indexposition der jeweiligen Anzahl der Exemplare im Array `anzahl`. Wird die Bauteilbestellung eines Kunden erfolgreich im System angelegt, so wird `true` als Rückgabewert zurückgegeben, andernfalls `false`.
Modellieren Sie in Material 2 die Methode `bauteileBestellen(…)` in einem UML-Sequenzdiagramm unter Berücksichtigung des UML-Klassendiagramms (Material 1).

(10 BE)

2 Die über die Buchungssoftware bestellten Bauteile werden mithilfe eines Industrieroboters in einzelne Tüten verpackt. Der Drehwinkel des Industrieroboters wird über einen, mit einer Regelung ausgestatteten, Elektromotor gesteuert.
Solche positionsgeregelten Elektromotoren bestehen aus dem eigentlichen Elektromotor, einem Sensor zur Ermittlung der Drehposition (z. B. einem Potentiometer) und einer Regelelektronik. Mithilfe eines periodischen Signals wird dem Motor der Soll-Winkel mitgeteilt. Üblich ist dabei ein 50-Hz-Signal (Periodendauer: 20 ms). Ein kurzes High-Signal von einer Dauer von 1 ms bringt den Motor an seinen rechten Endanschlag und ein High-Signal von 2 ms an seinen linken Endanschlag. Zeiten dazwischen stellen den Motor auf einen entsprechenden Winkel. Die restliche Zeit bis zum Ende der Periodendauer liegt ein Low-Signal an. Währenddessen hält der geregelte Motor die Position. Dabei ist die spezifische Stellzeit des Servos zu berücksichtigen, der für eine Winkeländerung um 40 Grad eine Stellzeit von 0,1 s (5 Periodendauern) benötigt und dessen Wiederholfrequenz 50 Hz beträgt. Ihre Aufgabe besteht darin, einen Teil der Mikrocontroller-Steuerung zu entwickeln.

2.1 Über einen Port des von Ihnen im Unterricht verwendeten Mikrocontrollers werden die Anzahl der Wiederholungen (entspricht der Anzahl der bestellten Exemplare von einem Bauteil) als 8-stellige Dualzahl eingelesen. Das erste Exemplar wird nun mithilfe des gegebenen Unterprogramms „Bauteil greifen“ von dem Industrieroboter gegriffen. Ein Sensor 1 erkennt, sobald das Bauteil gegriffen wurde, und liefert ein High-Signal (vorher Low-Signal). Sobald der Sensor 1 ein High-Signal liefert, soll der Industrieroboter um 40 Grad nach rechts drehen. Die dafür erforderliche High-Zeit beträgt 1,75 ms. Anschließend wird mithilfe des gegebenen Unterprogramms „Befüllen“ das Bauteil in eine Tüte verpackt. Ein Sensor 2 erkennt, wann der

Befüllvorgang abgeschlossen ist, und liefert ein High-Signal (vorher Low-Signal). Sobald der Sensor 2 ein High-Signal liefert, soll der Industrieroboter um 40 Grad zurück nach links drehen. Die dafür erforderliche High-Zeit beträgt 1,5 ms. Nun wird der Prozess für das nächste Exemplar wiederholt bis sämtliche Exemplare eines Bauteils in die Tüte gesteckt wurden und der Industrieroboter wieder in seine Anfangsposition zurückgedreht ist.
Stellen Sie die beschriebene Mikrocontroller-Steuerung für den Industrieroboter in einem Struktogramm dar.

Hinweis: Sämtliche erforderliche Verzögerungsschleifen können als gegeben angenommen und als Unterprogramme (z.B. warten1_5ms, warten1_75ms, etc.) verwendet werden.

(12 BE)

2.2 Ordnen Sie den Portpins des von Ihnen im Unterricht verwendeten Mikrocontrollers die Eingangs- und Ausgangsgrößen zu.

(2 BE)

2.3 Implementieren Sie die Mikrocontroller-Steuerung aus Aufgabe 2 in Assembler.

Hinweis: Die als Unterprogramme zur Verfügung gestellten Verzögerungsschleifen sowie die Programme „Bauteil greifen“ und „Befüllen“ müssen nicht implementiert werden.

(13 BE)

3 Die Information über die Anzahl der bestellten Exemplare eines Bauteils wird seriell von einem PC an den Mikrocontroller übertragen. Um diesen Wert an einem Port des Mikrocontrollers parallel einlesen zu können, wird ein Schieberegister benötigt.

3.1 Zeichnen Sie mithilfe des Datenblatts für die J-K-Flipflops vom Typ M74HC107 (Material 4) sowie des Datenblatts für die NAND-Gatter vom Typ 74HC00 (Material 5) den Verdrahtungsplan eines vierstufigen Schieberegisters mit serieller Eingabe und paralleler Ausgabe in die Vorlage in Material 3.

Hinweis: Sämtliche integrierten Bausteine sind mit einer Versorgungsspannung von 5 V zu betreiben.

(12 BE)

3.2 Zeichnen Sie unter Berücksichtigung der gegebenen Eingangs- und Taktsignale im Signal-Zeit-diagramm (Material 6) die Signale für die Ausgänge Q1 und Q2 eines vierstufigen Schieberegisters mit serieller Eingabe und paralleler Ausgabe unter der Annahme, dass im Schieberegister zum Zeitpunkt t=0 kein Signal gespeichert ist.

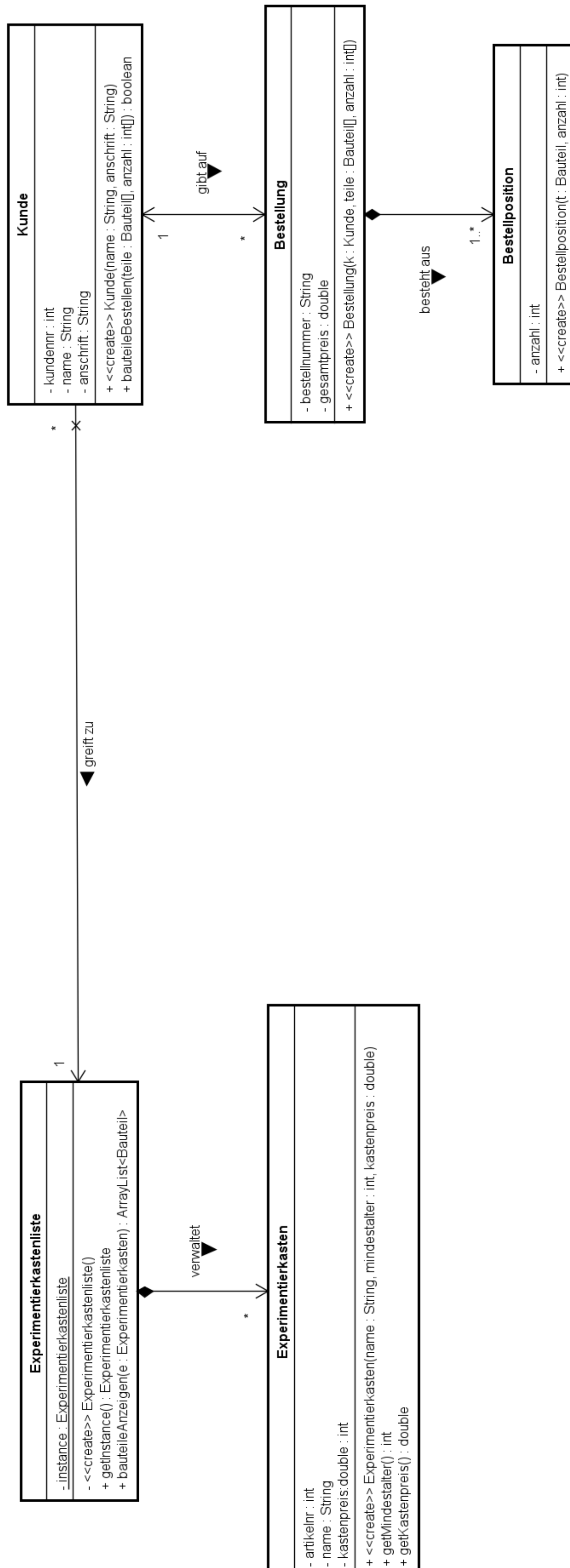
(6 BE)

3.3 Beschreiben Sie den Unterschied zwischen seriellem Laden, parallelem Laden sowie serieller Ausgabe und paralleler Ausgabe.

(4 BE)

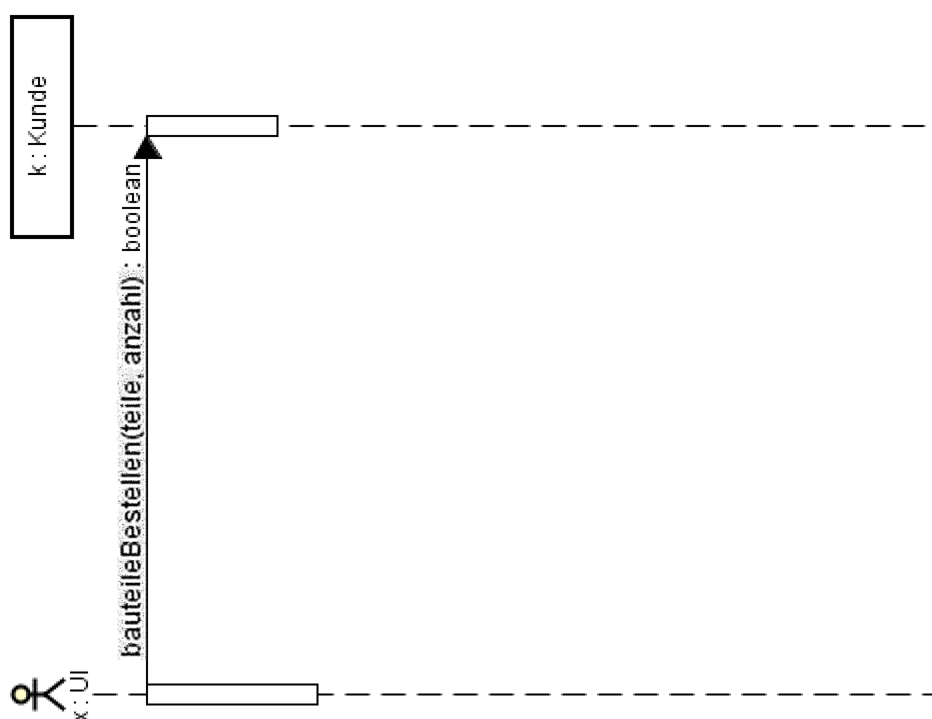
Material 1

UML-Klassendiagramm



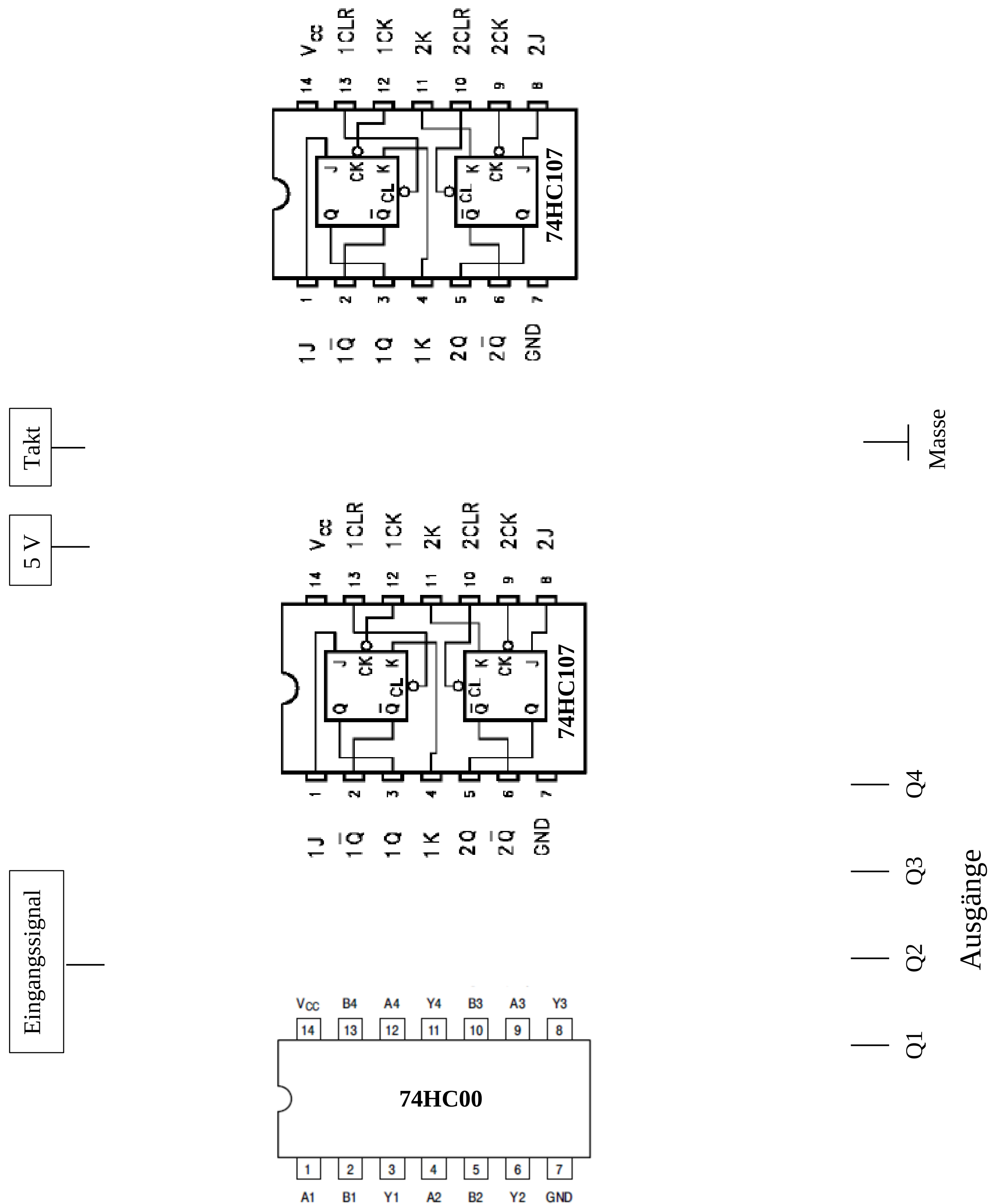
Material 2

UML-Sequenzdiagramm



Material 3

Vorlage Verdrahtungsplan des Schieberegisters



Material 4

Auszug aus Datenblatt der J-K-Flipflops vom Typ M74HC107



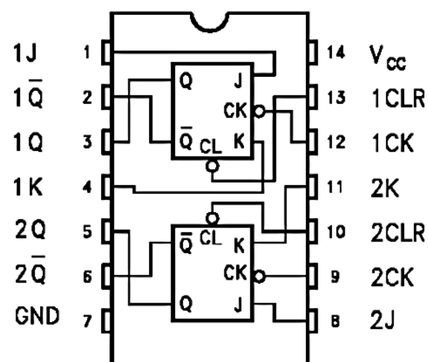
M54HC107
M74HC107

DUAL J-K FLIP FLOP WITH CLEAR

DESCRIPTION

The M54/74HC107 is a high speed CMOS DUAL J-K FLIP FLOP fabricated in silicon gate C^2MOS technology. It has the same high speed performance of LSTTL combined with true CMOS low power consumption. These flip-flop are edge sensitive to the clock input and change state on the negative going transition of the clock pulse. Each one has independent J, K, CLOCK, and CLEAR input and Q and $\overline{Q}$ outputs. CLEAR is independent of the clock and accomplished by a logic low on the input. All inputs are equipped with protection circuits against static discharge and transient excess voltage.

PIN CONNECTIONS (top view)



TRUTH TABLE

INPUTS				OUTPUTS		FUNCTION
$\overline{CLR}$	J	K	CK	Q	$\overline{Q}$	
L	X	X	X	L	H	CLEAR
H	L	L	↓	Q_n	$\overline{Q_n}$	NO CHANGE
H	L	H	↓	L	H	
H	H	L	↓	H	L	
H	H	H	↓	$\overline{Q_n}$	Q_n	TOGGLE
H	X	X	↑	Q_n	$\overline{Q_n}$	NO CHANGE

X: Don't Care

PIN DESCRIPTION

PIN No	SYMBOL	NAME AND FUNCTION
1, 8, 4, 11	1J, 2J, 1K, 2K	Synchronous Inputs; Flip-Flop 1 And 2
2, 6	$1\overline{Q}$, $2\overline{Q}$	Complement Flip-Flop Outputs
3, 5	1Q, 2Q	True Flip-Flop Outputs
12, 9	$1\overline{CK}$, $2\overline{CK}$	Clock Input
13, 10	1CLR, 2CLR	Asynchronous Reset Inputs
7	GND	Ground (0V)
14	V_{CC}	Positive Supply Voltage

http://pdf.datasheetcatalog.com/datasheet/SGSThomsonMicroelectronics/mXrvwsu.pdf (abgerufen am 13.03.2020).

Material 5

Auszug aus Datenblatt der NAND-Gatter vom Typ 74HC00

74HC00

Quad 2-Input NAND Gate

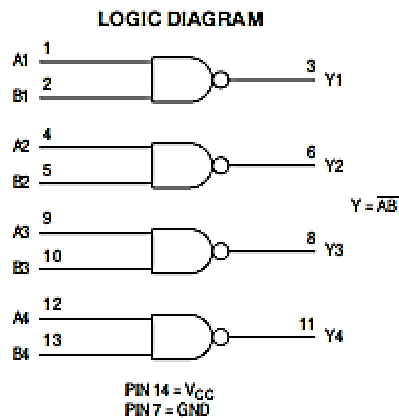
High-Performance Silicon-Gate CMOS

The 74HC00 is identical in pinout to the LS00. The device inputs are compatible with Standard CMOS outputs; with pullup resistors, they are compatible with LSTTL outputs.

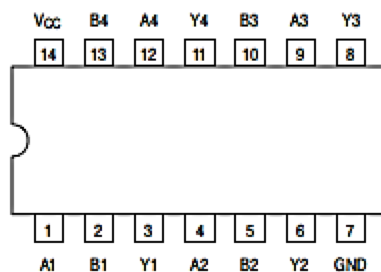
Features

- Output Drive Capability: 10 LSTTL Loads
- Outputs Directly Interface to CMOS, NMOS and TTL
- Operating Voltage Range: 2.0 to 6.0 V
- Low Input Current: 1.0 μA
- High Noise Immunity Characteristic of CMOS Devices
- In Compliance With the JEDEC Standard No. 7A Requirements
- ESD Performance: HBM > 2000 V; Machine Model > 200 V
- Chip Complexity: 32 FETs or 8 Equivalent Gates
- These are Pb-Free Devices

LOGIC DIAGRAM



Pinout: 14-Lead Packages (Top View)

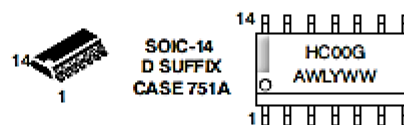


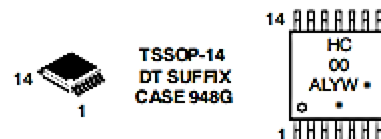


ON Semiconductor®

http://onsemi.com

MARKING DIAGRAMS





HC00 = Device Code
A = Assembly Location
WL or L = Wafer Lot
Y = Year
WW or W = Work Week
G or • = Pb-Free Package
(Note: Microdot may be in either location)

FUNCTION TABLE

Inputs		Output
A	B	Y
L	L	H
L	H	H
H	L	H
H	H	L

http://pdf.datasheetcatalog.com/datasheet/on_semiconductor2/74HC00.PDF (abgerufen am 19.09.2019).

Material 6

Signal-Zeit-Diagramm des Schieberegisters

